Experiential Solving: Towards a Unified Autonomous Search Constraint Solving Approach

Broderick Crawford[1,2,3](✉), Ricardo Soto[1,4,5], Kathleen Crawford[1], Franklin Johnson[1,6], Claudio León de la Barra[1], and Sergio Galdames[1]

[1] Pontificia Universidad Católica de Valparaíso, Valparaiso, Chile
{broderick.crawford,ricardo.soto,claudio.leondelabarra}@ucv.cl
kathleen.crawford.a@mail.pucv.cl,
sergiogaldames@gmail.com
[2] Universidad Central de Chile, Santiago, Chile
[3] Universidad San Sebastián, Santiago, Chile
[4] Universidad Autónoma de Chile, Santiago, Chile
[5] Universidad Cientifica del Sur, Lima, Peru
[6] Universidad de Playa Ancha, Valparaiso, Chile
franklin.johnson@upla.cl

Abstract. To solve many problems modeled as Constraint Satisfaction Problems there are no known efficient algorithms. The specialized literature offers a variety of solvers, which have shown good performance. Nevertheless, despite the efforts of the scientific community in developing new strategies, there is no algorithm that is the best for all possible situations. This paper analyses recent developments of Autonomous Search Constraint Solving Systems. Showing that the design of the most efficient and recent solvers is very close to the Experiential Learning Cycle from organizational psychology.

Keywords: Experiential learning · Problem solving · Metaheuristics · Autonomous search

1 Introduction

In constraint solvers development projects a better understanding of the human learning phenomenon offers important insights in order to obtain more efficient algorithms and therefore better problem solutions. By constraint solver, we mean the computer implementation of an algorithm for solving Constraint Satisfaction (and optimization) Problems (CSPs) [9]. A variety of approaches can be used to tackle CSPs. Integer programming techniques and constraint programming can be applied to find exact solutions. On the other hand, there are various approaches that provide an approximate solution, including metaheuristics and neural networks [2].

© Springer International Publishing Switzerland 2015
C. Stephanidis (Ed.): HCII 2015 Posters, Part I, CCIS 528, pp. 573–577, 2015.
DOI: 10.1007/978-3-319-21380-4_97

Since the functioning of modern constraint solvers is a learning intensive activity, an understanding of its operation from a learning phenomenon perspective can provide a valuable contribution for designing and implementing optimization algorithms in general and metaheuristics in particular. In this paper we present some basal ideas and concepts about learning and intelligence related with our work of structuring novelty solvers.

2 Explaining Learning: The Learning Cycle of Kolb

Kolb [6,7] developed a theory of experiential learning that can give us a useful model by which to develop better constraint solvers. The Learning Cycle or The Experiential Learning Cycle, as shown in Algorithm 1, comprises four different stages of learning from experience.

Algorithm 1. The Experiential Learning Cycle

```
while stop criteria is not satisfied do
    concrete experience (DOING)
    reflective observation (REVIEWING)
    abstract conceptualization (CONCLUDING)
    active experimentation (PLANNING)
end while
```

- Concrete Experience: doing/having an experience. A new experience of situation is encountered, or a reinterpretation of existing experience.
- Reflective Observation: reviewing/reflecting on the experience. Of particular importance are any inconsistencies between experience and understanding.
- Abstract Conceptualisation: concluding/learning from the experience. Reflection gives rise to a new idea, or a modification of an existing concept.
- Active Experimentation: planning/trying out what you have learned. The learner applies them to the world around them to see what results.

The Learning Cycle suggests that it is not sufficient to have an experience in order to learn. It is necessary to reflect on the experience to make generalisations and formulate concepts which can then be applied to new situations. This learning must then be tested out in new situations.

2.1 Linking Experiential Learning and Metaheuristics

As shown in Algorithm 2, the problem solving method used by metaheuristics presents a similar structure and operation to the cycle of Kolb. Subsequently, we are conducting study and research to discover the opportunities to improve our solvers through a better understanding of the learning phenomenon described by Kolb and others authors.

Algorithm 2. The Problem Solving Method of Metaheuristics

```
construct initial solutions (≃ DOING)
evaluate solutions (≃ REVIEWING)
rank solutions (≃ CONCLUDING)
select best solutions (≃ PLANNING)
while stop criteria is not satisfied do
    apply the metaheuristics operators to produce new solutions (≃ DOING)
    evaluate solutions (≃ REVIEWING)
    rank solutions (≃ CONCLUDING)
    select best solutions (≃ PLANNING)
end while
```

3 Explaining Intelligence: The Triarchic Theory of Sternberg

According to Sternberg [11], a complete explanation of intelligence entails the interaction of three subtheories:

- Componential subtheory which outlines the structures and mechanisms that underlie intelligent behavior categorized as metacognitive, performance, or knowlege acquistion components.
- Experiential subtheory that proposes intelligent behavior be interpreted along a continuum of experience from novel to highly familar tasks/situations.
- Contextual subtheory which specifies that intelligent behavior is defined by the sociocultural context in which it takes place and involves adaptation to the environment, selection of better environments, and shaping of the present environment.

In relation with the contextual subtheory, also called practical intelligence, it can be seen as an important referent to model adaptive constraint solvers. Following the principles underlying this subtheory recently it was introduced a new category of systems: Autonomous Search (AS) Systems [4,5].

3.1 Autonomous Search Systems

An autonomous search system should provide the ability to modify its internal components (heuristics, inference mechanisms, operators, movements, value parameters ...) when exposed to changing external forces and opportunities. As corresponds to an instance of adaptive systems with the objective of improving its problem solving performance by adapting its search strategy to the problem at hand. Autonomous search is particularly relevant to the constraint solving community, where much work has been conducted to improve the efficiency of constraint solvers. AS provides to a system the ability to change its components in order to improve its problem solving performance. AS can be defined as search processes that integrate control in their solving process either by self adaptation

or by supervised adaptation. This control allows an AS system to improve its solving performance by modifying and adjusting itself to the problem at hand. The notion of control is present when the parameters or heuristics are adjusted online, i.e., when the constraint solver is running. Different methods such as control encoding, control variable and value selection, and evolving heuristics have been proposed to provide control during solving.

Concerning the control, in self adaptation, techniques are tightly integrated with the search process and usually require some overhead. The algorithm is observing its own behavior in an online fashion, modifying its parameters accordingly. This information can be either directly collected on the problem or indirectly computed through the perceived efficiency of individual components. Because the adaptation is done online, there is an important trade-off between the time spent computing process information and the gains that are to be expected from this information.

The pioneer framework for AS in Constraint Programming was proposed in [1]. This approach was explainded in more details in [8] and it was applied successfully in [3,9]. The framework for AS can be seen as a 4-component architecture (see Algorithm 3): solve, observation, analysis and update.

- The *solve* component carries out the CSP resolution. The strategies employed in the process are selected from a ranked porfolio.
- *Observation* is responsible for taking and recording snapshots, which correspond to relevant information of the solving process.
- *Analysis* process the snapshots captured by *observation*. These snapshots are used to evaluate the strategies, which are stored in a database to be then gathered by *update*.
- *Update* is responsible for organizing the strategy rank.

Algorithm 3. Autonomous Search General Framework

```
while stop criteria is not satisfied do
    solve (≃ DOING)
    observation (≃ REVIEWING)
    analysis (≃ CONCLUDING)
    update (≃ PLANNING)
end while
```

Here, we can see the presence of the same operating structure proposed again in the cycle of Kolb.

4 Conclusions

Since the functioning of constraint solvers is a learning intensive activity, an understanding of its operation from a learning phenomenon perspective offers

important insights for designing and implementing better optimization algorithms and metaheuristics. In this paper we presented some ideas, concepts and experiences related with our work of structuring constraint solvers from a novelty point of view. It is clear that formalization of these influences is an area of research that is currently under-explored.

Acknowledgments. Broderick Crawford is supported by Grant CONICYT / FONDECYT / REGULAR / 1140897. Ricardo Soto is supported by Grant CONICYT / FONDECYT / INICIACION / 11130459.

References

1. Castro, C., Monfroy, E., Figueroa, C., Meneses, R.: An approach for dynamic split strategies in constraint solving. In: Gelbukh, A., de Albornoz, A., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 162–174. Springer, Heidelberg (2005)
2. Crawford, B., Soto, R., de la Barra, C.L., Crawford, K., Paredes, F., Johnson, F.: A better understanding of the behaviour of metaheuristics: a psychological view. In: Stephanidis [10], pp. 515–518
3. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredes, F.: Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. Expert Syst. Appl. **40**(5), 1690–1695 (2013)
4. Crawford, B., Soto, R., Olivares, R., Herrera, R., Monfroy, E., Paredes, F.: Autonomous search: towards the easy tuning of constraint programming solvers. In: Stephanidis [10], pp. 165–168
5. Hamadi, Y., Monfroy, E., Saubion, F.: Autonomous Search. Springer, Heidelberg (2012)
6. Kolb, D.: Chapter 15 - the process of experiential learning. In: Cross, R.L., Israelit, S.B. (eds.) Strategic Learning in a Knowledge Economy, pp. 313–331. Butterworth-Heinemann, Boston (2000)
7. Kolb, D.A.: Experiential Learning: Experience as the Source of Learning and Development. Prentice-Hall P T R, Englewood Cliffs (1984)
8. Monfroy, E., Castro, C., Crawford, B., Soto, R., Paredes, F., Figueroa, C.: A reactive and hybrid constraint solver. J. Exp. Theor. Artif. Intell. **25**(1), 1–22 (2013)
9. Soto, R., Crawford, B., Palma, W., Galleguillos, K., Castro, C., Monfroy, E., Johnson, F., Paredes, F.: Boosting autonomous search for CSPs via skylines. Inf. Sci. **308**, 38–48 (2015)
10. Stephanidis, C. (ed.): HCI 2014, Part I. CCIS, vol. 434. Springer, Heidelberg (2014)
11. Robert, J.: Sternberg. A triarchic approach to the understanding and assessment of intelligence in multicultural populations. J. Sch. Psychol. **37**(2), 145–159 (1999)